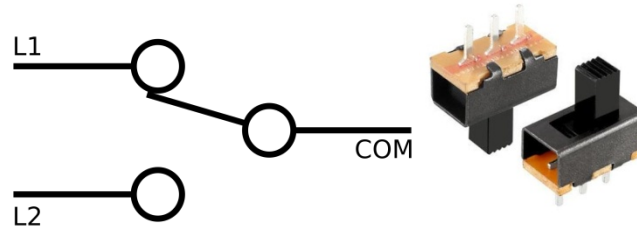


Uygulama 2: Anahtarlar ve Ledler.

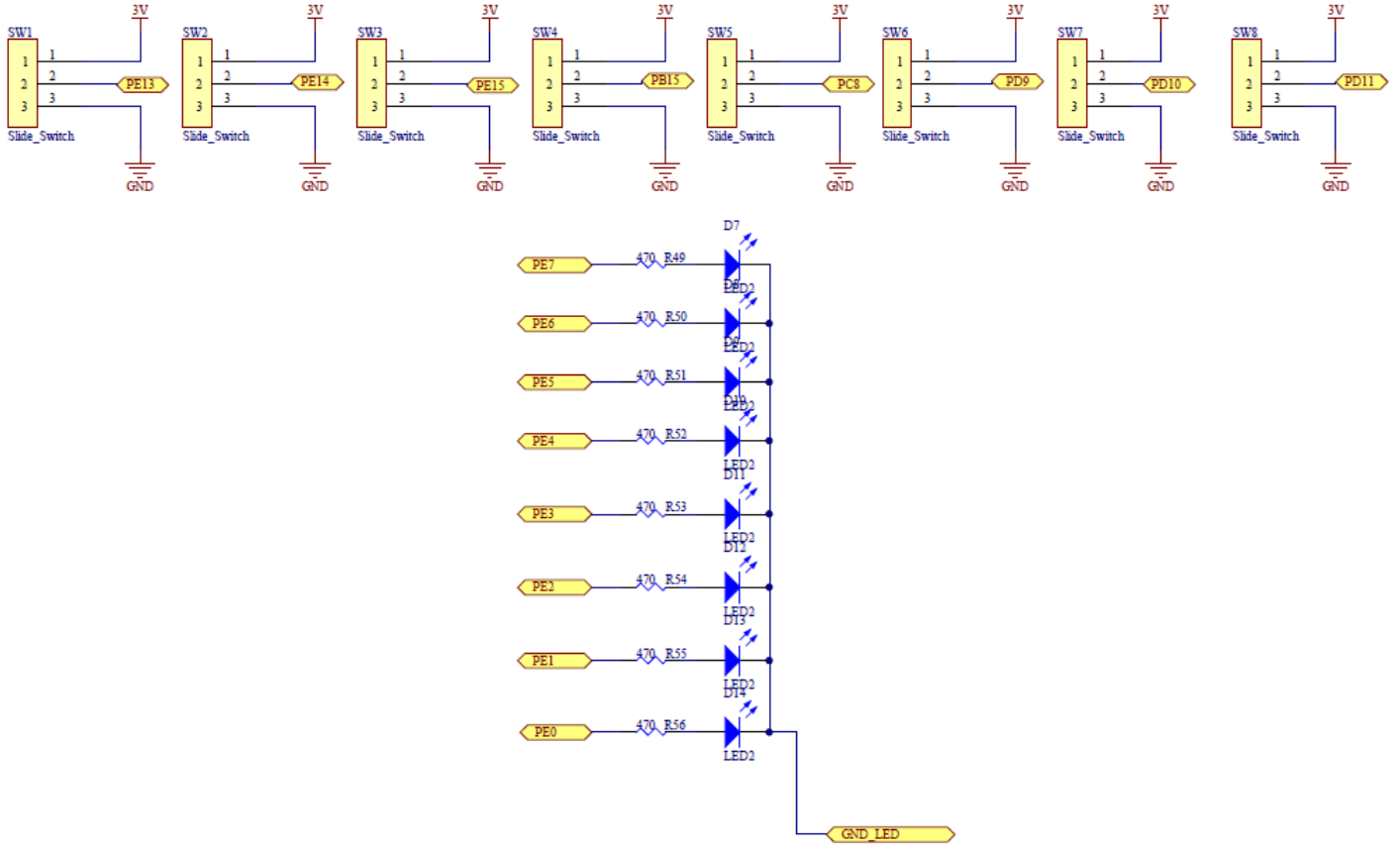
Anahtarlar elektronik devrelerde en sık kullanılan devre elemanlarındanlardır. Anahtarlar pek çok türde üretilmektedirler. ARMapp-18 uygulama setinde kullanılan anahtar tek kutup iki atımlı bir anahtardır. Bu tip anahtarlar 3 pinli olarak imal edilirler vs “SPDT” olarak adlandırılırlar.

Bu tip anahtarların ortada bulunan pinleri ortak uç, diğer ikisi ise birbirlerinden bağımsız uçlardır. Switch bir konuma alındığında Com ve L1 pinleri, diğer konuma alındığında ise Com ve L2 pinleri kısa devre olmaktadır.



Şekil 2.1: 3 pinli anahtar sembolü ve görüntüsü.

Deney setinde bu şekilde bir anahtar tercih edilmesinin sebebi bir konumda orta uca 3V, diğer konumda orta uca 0V uygulama ihtiyacını karşılayabilmesidir. Şekil 2.2’de bu uygulamada kullanılacak olan ledler ve anahtarlara ait bölümlerin ARMapp-18 deney setindeki devre şemaları görülebilir.



Şekil 3.2: ARMapp-18 Uygulama setinde bulunan Anahtar ve Led modüllerine ait şema.

Bu uygulamada anahtarların durumlarına göre kendilerine gelen ledler yakılacaktır. Örneğin SW7 orta ucundan 3V çıkacak konuma alındığında LED7 yanacak, orta ucundan 0V çıkacak konuma alındığında başka bir deyişle orta uç şaseye çekildiğinde LED7 sönecektir.

Uygulama Kodları:

```

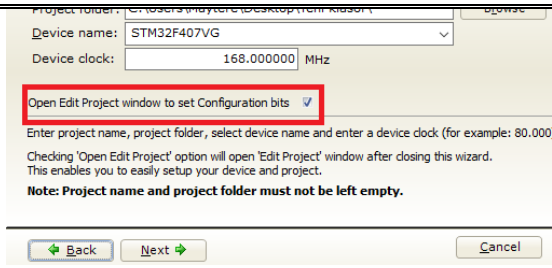
1 ////////////////////////////////////////////////////////////////////
2 //*****//
3 // Anahtarların ve Ledlerin kullanımı. //
4 // MikroC v6.2 - STM32F407VG //
5 // ARMapp-18 Deney Seti için yazılmıştır //
6 //*****//
7 // http://elektrovadi.com //
8 // http://mikrodunya.wordpress.com //
9 ////////////////////////////////////////////////////////////////////
10
11 void main() // Ana Program Bloğu
12 {
13     GPIO_Digital_Output(&GPIOE_BASE, _GPIO_PINMASK_0| // 'E' Portunun 0-7. pinleri dijital
14     _GPIO_PINMASK_1| // çıkış olarak ayarlanıyor.
15     _GPIO_PINMASK_2|
16     _GPIO_PINMASK_3|
17     _GPIO_PINMASK_4|
18     _GPIO_PINMASK_5|
19     _GPIO_PINMASK_6|
20     _GPIO_PINMASK_7);
21
22     GPIO_Digital_Input(&GPIOE_BASE, _GPIO_PINMASK_13| // 'E' Portunun 13,14 ve 15. pinleri dijital
23     _GPIO_PINMASK_14| // giriş olarak ayarlanıyor.
24     _GPIO_PINMASK_15);
25
26     GPIO_Digital_Input(&GPIOB_BASE, _GPIO_PINMASK_15); // 'B' Portunun 15. pini dijital
27     // giriş olarak ayarlanıyor.
28
29     GPIO_Digital_Input(&GPIOC_BASE, _GPIO_PINMASK_8 ); // 'C' Portunun 8. pini dijital
30     // giriş olarak ayarlanıyor.
31
32     GPIO_Digital_Input(&GPIOD_BASE, _GPIO_PINMASK_9 | // 'D' Portunun 9,10 ve 11. pinleri dijital
33     _GPIO_PINMASK_10| // giriş olarak ayarlanıyor.
34     _GPIO_PINMASK_11);
35
36     while(1) // Sonsuz döngü
37     {
38         char switches; // char tipinde 'switches' isminde bir değişken oluşturuluyor.
39         switches.b0=GPIOE_IDR.B13; // 'E' portunun 13. pininin değeri okunarak
40         // 'switches' değişkeninin 0. bitine aktarılıyor.
41         switches.b1=GPIOE_IDR.B14; // 'E' portunun 14. pininin değeri okunarak
42         // 'switches' değişkeninin 1. bitine aktarılıyor.
43         switches.b2=GPIOE_IDR.B15; // 'E' portunun 15. pininin değeri okunarak
44         // 'switches' değişkeninin 2. bitine aktarılıyor.
45         switches.b3=GPIOB_IDR.B15; // 'B' portunun 15. pininin değeri okunarak
46         // 'switches' değişkeninin 3. bitine aktarılıyor.
47         switches.b4=GPIOC_IDR.B8 ; // 'C' portunun 18. pininin değeri okunarak
48         // 'switches' değişkeninin 4. bitine aktarılıyor.
49         switches.b5=GPIOD_IDR.B9 ; // 'D' portunun 9. pininin değeri okunarak
50         // 'switches' değişkeninin 5. bitine aktarılıyor.
51         switches.b6=GPIOD_IDR.B10; // 'D' portunun 10. pininin değeri okunarak
52         // 'switches' değişkeninin 6. bitine aktarılıyor.
53         switches.b7=GPIOD_IDR.B11; // 'D' portunun 11. pininin değeri okunarak
54         // 'switches' değişkeninin 7. bitine aktarılıyor.
55         GPIOE_ODR=switches; // 'E' portuna switches değişkeni aktarılarak anahtarların
56         // durumu ledlerde gösteriliyor.
57     }
58 }
59
60
..



```

Kod kısmında switches isimli 8 bitlik bir değişken tanımlanmış ve bu değişkenin bitlerine anahtarların aldığı değerler okunarak aktarılmıştır. Daha sonra E portuna switches değişkeni olduğu gibi aktarılmış ve anahtarların konumuna göre ledlerin yanması sağlanmıştır.

İşlem Basamakları:

- 1- MikroC Pro for ARM programını çalıştırınız.
- 2- “Project” sekmesinden “New Project” seçeneğine tıklayınız.
- 3- Açılan ekrandan “Standart Project” seçilerek “Next” butonuna tıklayınız.
- 4- Açılan pencereden proje ismi, projenin kaydedileceği dizin, kullanılacak mikrodenetleyiciyi (STM32F407VG) ve mikrodenetleyici frekansını (168.000000 MHZ) belirleyerek ve “Open Edit Project window to set Configuration bits” seçeneğini işaretleyerek “Next” butonuna tıklayınız.



- 5- Açılan pencerede “Finish” butonuna tıklayarak proje oluşturma aşamasını bitiriniz.
- 6- Açılan pencerede sağ tarafta bulunan Load Scheme butonuna tıklayınız ve açılan pencereden “HSI 16 MHz_PLL 168 MHz Clock Ayarı.cfgsch” dosyasını seçiniz.
- 7- “OK” butonuna tıklayarak proje oluşturma işlemini tamamlayınız.
- 8- Açılan “.c” uzantılı sayfada kodları yazınız ve  ikonuna tıklayarak projeyi kaydediniz.
- 9- ARMap-18 setini USB kablo kullanarak bilgisayara bağlayınız.
- 10- Pişano anahtarlardan sol taraftakinde bulunan LED anahtarını açarak, ledlerin GND bağlantısını sağlayınız.
- 11-  “Build and Program” ikonuna tıklanarak veya (Ctrl+F11) tuşlarına aynı anda basarak yazılan programı derleyip STM32F407VG mikrodenetleyicisine yükleyiniz.
- 12- Programın çalışmasını anahtarların konumlarını değiştirerek gözlemleyiniz.

Sorular:

- 1- Anahtarlar şu anda 1 konumuna alındığında ledler yanmaktadır. Ledlerin anahtarlar 0 konumuna alındığında yanmasını sağlayınız.
- 2- SW7 - LED0
SW6 - LED1
SW5 - LED2
SW4 - LED3
SW3 - LED4
SW2 - LED5
SW1 - LED6
SW0 - LED7
eşleşmesi ile anahtarların eşleştikleri ledlerin durumlarını kontrol etmelerini sağlayınız.

Uygulama 3: Menü Tuş Takımı.